**Bước 1: Sửa class LoginResponse ở backend**

Thêm trường admin:

|  |
| --- |
| public class LoginResponse {  private String token;  private AdminUser admin;  public LoginResponse(String token, AdminUser admin) {  this.token = token;  this.admin = admin;  }  public String getToken() { return token; }  public AdminUser getAdmin() { return admin; }  } |

**Bước 2: Sửa AuthController trả về cả token và admin**

|  |
| --- |
| @PostMapping("/login")  public ResponseEntity<?> login(@RequestBody LoginRequest request) {  AdminUser user = adminUserRepository.findByUsername(request.getUsername())  .orElseThrow(() -> new RuntimeException("Tài khoản không tồn tại"));  if (!user.getPassword().equals(request.getPassword())) {  return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Sai mật khẩu");  }  String token = jwtUtil.generateToken(user.getUsername());  return ResponseEntity.ok(new LoginResponse(token, user));  } |

Thêm thư viện

|  |
| --- |
| <dependency>  <groupId>com.mysql</groupId>  <artifactId>mysql-connector-j</artifactId>  <version>8.0.33</version>  <scope>runtime</scope>  </dependency> |

Cấu hình application.properties

|  |
| --- |
| spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver |

# **1. Cập nhật thông tin admin**

**Thêm vào AdminUserController:**

|  |
| --- |
| @PutMapping("/{id}")  public AdminUser updateInfo(@PathVariable int id, @RequestBody AdminUser updated) {  AdminUser admin = service.getById(id);  if (admin == null) throw new RuntimeException("Admin not found");  // Cập nhật các trường cho phép  admin.setFullName(updated.getFullName());  admin.setEmail(updated.getEmail());  admin.setAvatarUrl(updated.getAvatarUrl());  return service.save(admin);  } |

# **2. Đổi mật khẩu admin**

**Tạo DTO cho đổi mật khẩu:**

|  |
| --- |
| // API/src/main/java/com/example/API/DTO/Request/ChangePasswordRequest.java  package com.example.API.DTO.Request;  import lombok.Data;  @Data  public class ChangePasswordRequest {  private String oldPassword;  private String newPassword;  } |

**Thêm vào AdminUserController:**

|  |
| --- |
| @PostMapping("/{id}/change-password")  public String changePassword(@PathVariable int id, @RequestBody ChangePasswordRequest req) {  AdminUser admin = service.getById(id);  if (admin == null) throw new RuntimeException("Admin not found");  if (!admin.getPassword().equals(req.getOldPassword())) {  return "Mật khẩu cũ không đúng!";  }  admin.setPassword(req.getNewPassword());  service.save(admin);  return "Đổi mật khẩu thành công!";  } |

**Nhớ import:**

|  |
| --- |
| import com.example.API.DTO.Request.ChangePasswordRequest; |

Thêm thư viện:

|  |
| --- |
| *implementation* ("com.github.bumptech.glide:glide:4.16.0") |

Code file ProfileFragment

Code file EditInfoFragment

Sửa thêm file ApiService

Thêm file SharedPreferencesManager

* **ProfileFragment**: Khi vào sẽ tự động gọi API lấy thông tin user theo id và hiển thị avatar, tên, email lên UI.
* **EditInfoFragment**: Khi vào cũng tự động gọi API lấy thông tin user theo id và hiển thị fullname, email, username vào các ô nhập liệu.

**Điều kiện:**

* Khi đăng nhập, bạn phải lưu id user vào SharedPreferences (đã sửa lại hàm saveAdminInfo).
* API backend /api/admin-users/{id} phải hoạt động đúng.